

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently amended) A method for use in each of two server computers to determine, when one of the server computers is restored from a fault, determining a server computer which whether the server computer carried out a process most recently, applicable to a high availability computer system in which when one of the server computers executes the process and a fault occurs in the server computer, the other server computer takes over the process, comprising a master server computer and a slave server computer each of the server computers having a storage unit, the method comprising the steps of:
executing state-transition of said two the server computers when a fault occurs in one of the server computers computer or one of the server computers computer is restored from the fault, each of the server computers being constructed to assume four states, which are master state in which the server computer carries out the process and has a mate which takes over the process, single master state in which the server computer carries out the process and has no mate which takes over the process, slave state in which the server computer does not carry out the process but has information necessary for taking over of the process, and halt state in which the server computer does not carry out the process and

holds no further information necessary for taking over of the process;

storing a priority determined by the state-transition into the storage unit[[;]],

the step of storing including the steps of:

changing the priority of the server computer so as to indicate

the highest priority when the state of the server

computer is changed to the single master state;

changing the priority of the server computer so as to indicate

the second highest priority when the state of the

server computer is changed to the master state;

changing the priority of the server computer so as to indicate

the lowest priority when the state of the server

computer is changed to the slave state; and

prohibiting a change of the priority of the server computer

when the state of the server computer is changed to

the halt state,

determining, when the server computer is restored from the fault, whether

~~or not the priority of the server computer restored from the fault is~~

~~higher; [[and]]~~

determining whether the server computer carried out the process most

recently, based on the result of determining whether the priority of

the server computer is higher; and

determining that the server computer ~~restored from the fault~~ becomes a server computer to take over [[a]] the process, when it is determined that the priority of the server computer is higher carried out the process most recently.

2. (Original) The method according to claim 1, wherein the step of determining the priority further including includes the step of:
comparing the priorities of the server computers when faults occur in the server computers and after that, the server computers are restored from the faults so as to determine which priority is higher.

3. (Currently amended) The method according to claim 2, wherein the step of determining the priority further including includes the steps of:
determining, when faults occur in the server computers and after that, only one thereof of the server computers is restored from the fault, whether ~~or not~~ the priority of the restored server computer ~~restored from the fault~~ is the highest priority; and
determining that the priority of the server computer is higher only when the priority is determined to be the highest priority.

4. (Cancelled)

5. (Cancelled)

6. (Currently amended) The method according to claim [[4]] 3, wherein the step of executing the state-transition further ~~including~~ includes the steps of:
- executing, when faults ~~occurs~~ occur in [[said]] ~~the~~ server computers and after that, [[said]] ~~the~~ server computers are restored from the faults, [[the]] a first state-transition such that one of [[said]] ~~the~~ server computers is changed from the halt state to the single master state while the other [[one]] server computer is changed from the halt state to the slave state, based on at least the result of determining whether that the given ~~the~~ server computer carried out the process most recently becomes a server computer to take over a process;
- and
- executing, after the first state-transition is completed, [[the]] a second state-transition such that [[the]] one server computer is changed to the master state while the other [[one]] server computer remains in the slave state.
- ;
7. (Currently amended) The method according to claim [[4]] 6, wherein the step of executing the state-transition further ~~including~~ includes the steps of:
- executing [[the]] a third state-transition such that ~~the server restored from the fault~~ one of the server computers is changed from the halt state to the single master state or keeping the current state, based on at least the result of determining whether that the server computer

carried out the process most recently restored from the faults
~~becomes a sever computer to take over a process, when faults~~
occur in [[said]] the server computers and after that, only any one
thereof of the server computers is restored from the fault; and
executing, when after the step of executing the third state-transition or
keeping the current state is completed, the other server computer
of [[said]] the server computers is also restored from the fault, either
[[the]] a fourth state-transition such that the other server computer
is changed from the halt state to the slave state or [[the]] a fifth
state-transition in which any one of [[said]] the server computers is
changed to the single master state while the other [[one]] server
computer is changed to the slave state.

8. (Currently amended) The method according to claim 3, wherein the step of executing the state-transition further including includes the step of:
executing, after faults occurs occur in [[said]] the server computers and
then, only [[any]] one of the server computers computer is restored
from the fault, the state-transition of the server computer such that
the server computer is a server computer which continues the
process, when the server computer is not capable of being
determined to have a priority higher than the other server computer,
because the priority stored in the storage unit of the server
computer restored from the fault is not the highest priority and then,

a forced start instruction for forcing the server computer to continue the process is given from outside.

9. (Currently amended) A high availability computer system comprising two server computers a master server computer and a slave server computer each having a storage unit, each of the [[said]] server computers comprising:

state-transition means for executing a state-transition of [[said]] the server computers when a fault occurs in the server computer one of the server computers or one of the server computers computer is restored from the fault, each of the server computers being constructed to assume four states, which are master state in which the server computer carries out a process and has a mate which takes over the process, single master state in which the server computer carries out the process and has no mate which takes over the process, slave state in which the server computer does not carry out the process but has information necessary for taking over of the process, and halt state in which the server computer does not carry out the process and holds no further information necessary for taking over of the process;

state writing means for storing a priority determined by the state-transition into the storage unit;

first determining means for determining, when the [[each]] server computer is restored from the fault, whether or not the priority of the [[each]] server computer is higher; [[and]]

second determining means for determining whether the server computer carried out the process most recently, based on the determination result of the first determining means; and

second third determining means for determining that the [[each]] server computer restored from the fault becomes a server computer to take over [[a]] the process, when it is determined that the server computer carried out the process most recently.

10. (Currently amended) A high availability computer system according to claim 9, wherein the first determining means compares the priorities of the server computers, when faults occur in the server computers and then the server computers are restored from the faults, so as to determine which priority is higher.

11. (Currently amended) A high availability computer system according to claim 10, wherein the first determining means, when faults ~~occurs~~ occur in the server computers and then only one of the [[each]] server computers ~~computer~~ is restored from the fault, determines whether or not the priority of the [[each]] server computer is the highest priority and only when the priority is the highest priority, determines that the priority of the [[each]] server computer is higher.

12. (Cancelled)

13. (Currently amended) A high availability computer system according to claim [[12]]

11, wherein the state-transition means, when faults occur in the server computers and then the server computers are restored from the faults, executes such state-transition that one of the server computers is changed from the halt state to the single master state while the other [[one]] server computer is changed from the halt state to the slave state, based on at least the determination result of the second third determining means, and then executes such state-transition that the [[one]] server computer is changed to the master state while the other [[one]] server computer remains in the slave state.

14. (Currently amended) A high availability computer system according to claim [[12]]

11, wherein the state-transition means, when faults occur in the [[said]] server computers and then only one of the [[each]] server computers ~~computer~~ is restored from the fault, either executes such state-transition that the [[each]] server computer is changed from the halt state to the single master state or keeps the current state, based on at least the determination result of the second third determining means, and then when the other server computer of the server computers is also restored from the fault, executes either state-transition that the other server computer is changed from the halt state to the slave state or state-transition in which any one of the server computers is changed to the single

master state while the other [[one]] server computer is changed to the slave state.

15. (Currently amended) A computer readable recording medium storing a program, executed by each of two server computers, for determining, when one of the server computers is restored from a fault, a server computer which whether the server computer carried out a process most recently, applicable to high availability computer system in which when one of the server computers executes the process and a fault occurs in the server computer, the other server computer takes over the process, comprising a master server computer and a slave server computer each server computer having a storage unit, the program comprising:

code means for when a fault occurs in one of the server computers computer or one of the server computers computer is restored from the fault, making causing the server computer to execute state-transition of [[said]] the server computers, each of the server computers being constructed to assume four states, which are master state in which the server computer carries out the process and has a mate which takes over the process, single master state in which the server computer carries out the process and has no mate which takes over the process, slave state in which the server computer does not carry out the process but has information necessary for taking over of the process, and halt state in which the

server computer does not carry out the process and holds no further information necessary for taking over of the process;

code means for ~~making~~ causing the server computer to store a priority determined by the state-transition into the storage unit[[;]], the code means for causing the server computer to store a priority including:

code means for causing the server computer to change the priority of the server computer so as to indicate the highest priority when the state of the server computer is changed to the single master state;

code means for causing the server computer to change the priority of the server computer so as to indicate the second highest priority when the state of the server computer is changed to the master state;

code means for causing the server computer to change the priority of the server computer so as to indicate the lowest priority when the state of the server computer is changed to the slave state; and

code means for causing the server computer to prohibit a change of the priority of the server computer when the state of the server computer is changed to the halt state.

code means for when the server computer is restored from the fault, making causing the server computer restored from the fault to

determine whether or not the priority of the server computer

~~restored from the fault~~ is higher; [[and]]

code means for causing the server computer to determine whether the
server computer carried out the process most recently, based on
the result of determining whether the priority of the server computer
is higher; and

code means for ~~making causing~~ the server computer ~~restored from the~~
~~fault to~~ determine that the server computer restored from the fault
becomes a server computer to take over [[a]] the process, when the
priority is higher it is determined that the server computer carried
out the process most recently.